

Dosbox.conf

From DOSBoxWiki

[Jump to navigation](#)[Jump to search](#)

dosbox.conf is a configuration file that DOSBox can use globally and/or locally per game (and settings that are left out are taken from the global file). It contains various system settings and initialization values that define your emulated environment. Everything can be controlled by editing this file or if you like through more graphically oriented Front Ends. You can also create separate **dosbox.conf** files for multiple host environments (which is helpful in playing various DOS games that expect various types of hardware).

The configuration file is broken into separate sections which contain section settings. Many of these settings do not need to be fully understood to configure DOSBox, but it is helpful to know where to look. You should also be aware that anything to the right of the # to the end of the line is considered a comment as is totally ignored by DOSBox when it loads.

Contents

- 1 Creation and Location
 - 1.1 Windows
 - 1.1.1 Windows XP
 - 1.1.2 Windows Vista, Windows 7 & Windows 8.x
 - 1.2 Linux
 - 1.3 Mac OS X
 - 1.4 Wine
- 2 Sections
 - 2.1 [sdl]
 - 2.2 [dosbox]
 - 2.3 [render]
 - 2.3.1 frameskip = *nnn*
 - 2.3.2 aspect = true | **false**
 - 2.3.3 scaler = none | normal2x | normal3x | tv2x | tv3x | rgb2x | rgb3x | scan2x | scan3x | advmame2x | advmame3x | advinterp2x | advinterp3x | 2xsai | super2xsai | supereagle | hq2x | hq3x
 - 2.4 [cpu]
 - 2.4.1 core = simple | normal | dynamic | auto
 - 2.4.2 cputype = auto | 386 | 386_slow | 486_slow | pentium_slow | 386_prefetch
 - 2.4.3 cycles = fixed *nnnn* | max [*default%*] [*limit cycle limit*] | auto [*realmode default*] [*protected mode default%*] [*limit "cycle limit"*]
 - 2.4.3.1 Examples:
 - 2.4.4 cycleup = *nnn*
 - 2.4.5 cycledown = *nnn*
 - 2.5 [mixer]
 - 2.6 [midi]
 - 2.6.1 mpu401 = intelligent | uart | none

- 2.6.2 mididevice = default | win32 | alsa | oss | coreaudio | coremidi | none
- 2.6.3 midiconfig = *id of MIDI device*
- 2.7 [sbblaster]
- 2.8 [gus]
- 2.9 [speaker]
- 2.10 [joystick]
- 2.11 [serial]
- 2.12 [dos]
- 2.13 [ipx]
- 2.14 [autoexec]
- 3 Recommendations

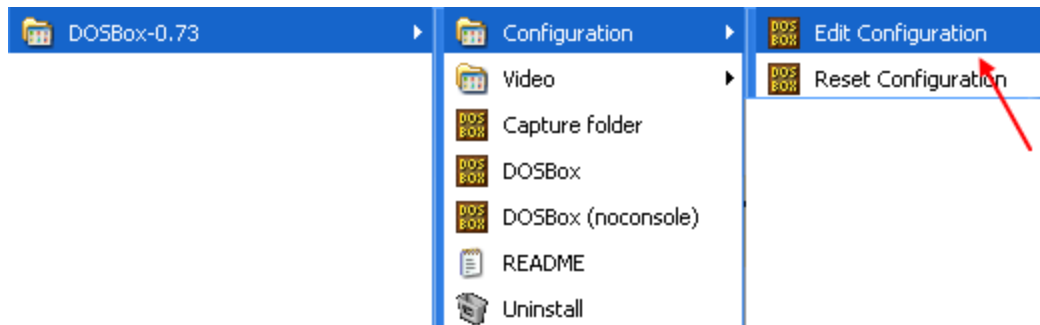
Creation and Location

Depending on the version or host OS, the dosbox.conf file is located either inside the user profile folder or inside the same folder as dosbox.exe. On 0.74, just go to the DOSBox folder and open "DOSBox 0.74 Options.bat" in order to open the configuration file (*.conf).

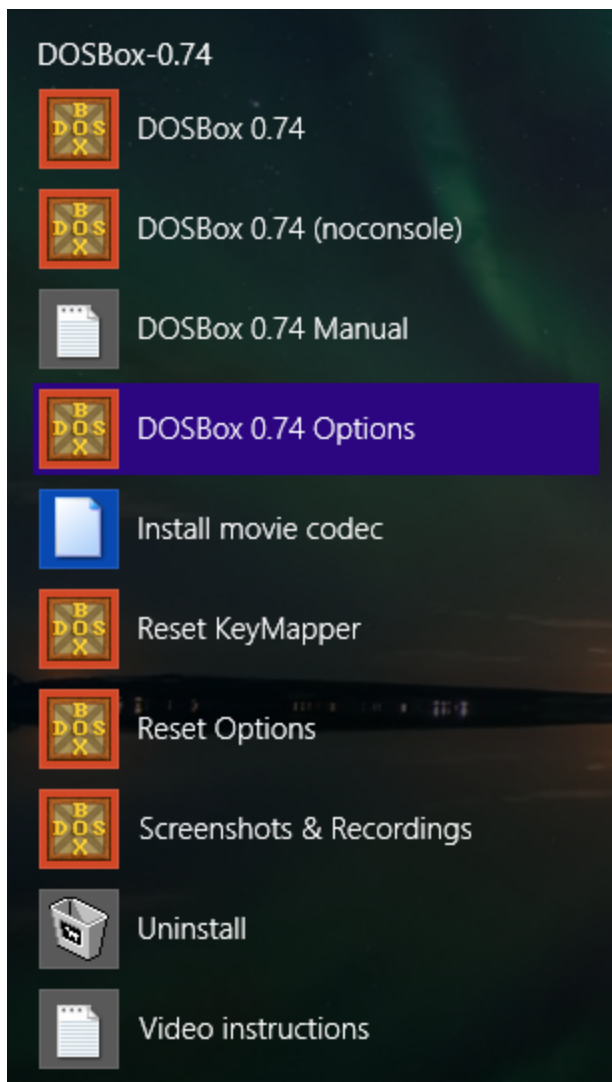
Windows

Dosbox.conf is created automatically in the Windows' user profile folder. The location is indicated by the DOSBox Status Window upon startup.

You can find the **dosbox.conf** by browsing **Start > All Programs > Dosbox-{version}**



Or:



Windows XP

%USERPROFILE%\Local Settings\Application Data\DOSBox\dosbox-{version}.conf

Windows Vista, Windows 7 & Windows 8.x

{system drive}:\Users\{username}\AppData\Local\DOSBox\dosbox-{version}.conf

Linux

For Linux the configfile is created on the first run in ~/.dosbox/
The name is dosbox-{version}.conf where version is currently 0.74

On Ubuntu in the dosbox man file it is written:

Configuration and language files use a format similar to Windows .ini files. First ~/.dosboxrc (if present) will be loaded. If no configfile is specified at the commandline, a file named dosbox.conf (if present in the current directory) will be loaded automatically after-

wards. If a configfile is specified at the commandline that one will be used instead.

Mac OS X

If you are using Mac OS X, a preferences file will be created for you on the first time you run DOSBox (as of version 0.73). This file contains the same system settings and initialization values as the **dosbox.conf** file on other systems.

It can be found (and modified) at `~/Library/Preferences/DOSBox 0.73 Preferences`, where `~/` is your user profile folder (usually `/Macintosh HD/Users/username/`). The exact folder name in the Finder ([http://en.wikipedia.org/wiki/Finder_\(software\)](http://en.wikipedia.org/wiki/Finder_(software))) may vary, depending on the language you use for OS X.

Wine

Wine makes a temporary config file every time it runs at `/.wine/drive_c/users/${USER}/Temp/cfg?????.tmp`. This file is also deleted after DOSBox is launched.

Sections

[sdl]

This section contains all of the low level system settings for how DOSBox interacts with your real hardware. You can define what resolutions are emulated, how DOSBox should treat errors or listen to your keyboard and mouse. You can often achieve a fair level of optimization by working with these setting, though for the most part leaving them at their default settings will create the best experience. These settings are passed on to the SDL Library which handles low level things like input and thread priority.

fullscreen = true | false

Start DOSBox directly in fullscreen.
Default is false.

fulldouble = true | false

Use double buffering in fullscreen. See [1] (http://en.wikipedia.org/wiki/Multiple_buffering)
Default is false.

fullresolution = *width x height* | original | desktop

Scale the application to this size IF the output device supports hardware scaling (i.e. any output other than surface). Original is the game's default or chosen (through setup.exe or in-game menu) resolution. If original resolution is less than desktop resolution, DOSBox will switch the screen resolution to the closest match requested by the game or application. For example, if a game in DOSBox is requesting a graphics screen resolution of (320 x 240) while your desktop is (1920 x 1200), DosBox will switch to (320x240) or the next highest resolution supported by your GPU drivers, e.g. (800 x 600) if the former is not available. Many games will be below the minimum resolution supported by modern video cards, so DOSBox will scale the game up to at least that minimum. Note: the scaler setting under [render] is also able to scale up the original resolution to some degree. Those changes are performed before any additional scaling done with fullresolution setting.
Default is original.

windowresolution = *width x height* | original | desktop

Scale the window to this size IF the output device supports hardware scaling (i.e. any output other than surface). Original is the game's default or chosen (through setup.exe or in-game menu) resolution. Note: the scaler setting under [render] is also able to scale up the original resolution to some degree. Those changes are performed before any scaling done with windowresolution setting.

Default is original.

output = surface | overlay | opengl | openglfb | ddraw

What to use for output. Surface does not support scaling or aspect correction. More information here: [2] (<http://www.vogons.org/viewtopic.php?=&p=102291>)

Default is surface.

autolock = true | false

Mouse will automatically lock, if you click on the screen.

Default is true.

sensitivity = 1..1000

Mouse sensitivity.

Default is 100.

waitonerror = true | false

Wait before closing the console if DOSBox has an error.

Default is true.

priority = *when-focused,when-minimized*

Priority levels for DOSBox. Second entry behind the comma is for when DOSBox is not focused/minimized. Valid priorities are: lowest, lower, normal, higher, highest, and pause.

Default is higher,normal.

mapperfile = *path-to-mapper-file*

File used to load/save the key/event mappings from.

Default is mapper-<version>.txt.

usescancodes = true | false

Avoid usage of symkeys, might not work on all operating systems.

Default is true.

[dosbox]

The **[dosbox]** section contains various settings that do not pertain to any other section (e.g. setting the language used in DOSBox help texts, where to store screen captures, etc.)

language = *path-to-language-file*

Select another language file.

The default value empty (*language=*).

(since 0.??)

memsize = nn

Amount of high memory (in megabytes) available to programs.

Note: DOSBox always allocates 1 MB of low memory, so the total amount of memory equals 1 MB of low memory, plus whatever is allocated for high memory.

The default value is 16 (*memsize=16*).
(since 0.??)

machine = **hercules** | **cga** | **tandy** | **cga** | **tandy** | **pcjr** | **ega** | **vgaonly** | **svga_s3** | **svga_et3000**
| **svga_et4000** | **svga_paradise** | **vesa_nolfb** | **vesa_oldvbe**

(since 0.73)

(previously **machine** = **hercules** | **cga** | **tandy** | **vga**)

The type of machine (specifically the type of graphics hardware) DOSBox tries to emulate.

The default value is **svga_s3** (*machine=svga_s3*) (was previously **vga**)

Definitions are as follows:

svga_s3 (**vga** on DOSBox 0.71 and 0.72) is *SVGA (Super Video Graphics Array)*: A loose standard designed to allow graphics modes superior to that of VGA. This option emulates an S3 Trio64, one of the most compatible SVGA cards, supporting 256 colors at up to 1600x1200 and full (32- or 24-bit) color at up to 1024x768. It is nearly 100% backwards compatible with VGA, and thus is backwards compatible with EGA and CGA (except 16-color composite mode). It is not backwards compatible with special Tandy, PCjr or Hercules Monochrome graphics modes.

(info taken from <http://ipggi.wordpress.com/2008/03/16/dosbox-graphic-and-machine-emulation-cga-vga-tandy-pcjr-hercules/> and <http://www.datasheetarchive.com/S3%20TRIO%2064-datasheet.html>)

vgaonly (**vga** on DOSBox 0.70 and previous) is *VGA (Video Graphics Array)*: IBM's graphics system introduced with the PS/2. True VGA supports 16 colors at 640x480 resolution, or 256 colors at 320x200 resolution (and not 256 colors at 640x480, even though many people think it does). VGA colors are chosen from a palette of 262,144 colors (not 16.7 million) because VGA uses 6 bits to specify each color, instead of the 8 that is the standard today.

(info taken from <http://www.pcguide.com/ref/video/stdVGA-c.html>. See also wikipedia:Video Graphics Array and <http://members.chello.at/theodor.lauppert/games/vga.htm>)

CGA (Color Graphics Adapter): Refers to IBM's first color graphics card. The CGA supports several different modes; the highest quality text mode is 80x25 characters in 16 colors. Graphics modes range from monochrome at 640x200 (which is worse than the Hercules card) to 16 colors at 160x200 in composite mode. However, for gaming, by far the most common mode was 4 colors at 320x200 pixels. While various hacks allowed substitution of one of the 16 colors above, there were only two official palettes for this mode:

- Magenta, cyan, white and background color (black by default). (much more common for gaming)
- Red, green, brown and background color (black by default). (Can sometimes be selected as an alternate on some games).

Note that SVGA fully supports most CGA modes (except the 16-color composite mode), so you should be able to leave DosBox in SVGA mode and play most CGA games.

(info taken from wikipedia:Color Graphics Adapter, <http://www.pcguide.com/ref/video/stdCGA-c.html>, and <http://members.chello.at/theodor.lauppert/games/cga.htm>)

Tandy: Refers to the additional graphics modes available on a Tandy 1000 or PCjr, which included 160x100x16, 160x200x16, 320x200x16, and 640x200x4. The Tandy

RL/SL/TL series also added a 640x200x16 mode. Also backwards compatible with CGA, except for 16-color composite mode.

(info taken from <http://www.mobymgames.com/attribute/sheet/attributeId,31/>)

PCjr: Refers not to a graphics mode but an entire system developed by IBM, and the company's first attempt at a home computer. Its graphics modes are identical to that of the Tandy, but it is not 100% compatible with any other IBM computer. Allows the user to boot cartridge files specifically designed for this system (.jrc).

(info from wikipedia:IBM PCjr and www.dosbox.com/DOSBoxManual.html)

Hercules: Refers to a graphics card developed by Hercules Computer Technology as a competitor to CGA for monochrome monitors. Hercules systems generate both high-resolution text and graphics. The resolution is 720 by 348 and only a two colors (foreground and background) are supported.

(info taken from http://www.webopedia.com/TERM/H/Hercules_graphics.html and wikipedia:Hercules Graphics Card)

EGA (*Enhanced Graphics Adapter*): Refers to a graphics card developed by IBM between the CGA and VGA, and thus has capabilities between the two. It supports 16 colors at a time from a set of 64 possible in resolutions up to 640x350. It also supports all the color modes of CGA except 16-color composite mode. This mode is rarely needed as VGA and SVGA can handle all the same modes.

(info taken from wikipedia:Enhanced Graphics Adapter)

Additional SVGA modes (only necessary if **svga_s3** doesn't work.)

- **vesa_oldvbe** the same as **svga_s3**, but uses a lower version of VESA (1.3), the code used to interface with SVGA cards. This is necessary for some older SVGA programs.
- **vesa_nolfb** the same as **svga_s3** introduces a no-line frame buffer hack. Sometimes runs faster than plain **svga_s3**. Only needed in a few games due to either a bug in DOSBox or the line-frame buffer mode of the game.
- **svga_paradise** emulates the Paradise PVGA1A and is only capable of the SVGA resolutions of 256 colors at 640×480 and 16 colors at 800×600. Like all SVGA modes, it is nearly 100% backwards compatible with VGA.
- **svga_et3000** emulates the Tseng Labs ET3000, which can handle everything the Paradise card can plus 256 colors at 800×600 and 16 colors at 1024×768.
- **svga_et4000** emulates the Tseng Labs ET4000, which is the same as the ET3000 except it can also handle 256 colors at 1024×768.

(info taken from <http://ipggi.wordpress.com/2008/03/16/dosbox-graphic-and-machine-emulation-cga-vga-tandy-pcjr-hercules/>)

captures = path-to-capture-directory

Directory where things like music (wave and MIDI) and screenshots are captured when special keys CTRL-F5 and CTRL-F6 are used. Screenshots will be captured and saved as (PNG) files with a resolution of 320x200.

Note: ~~The capture directory will not be created automatically - you must create it before you start capturing music and screenshots, otherwise nothing will be saved.~~ As of v0.73 (possibly prior) it is created automatically on first use.
The default value is capture (*captures=capture*).
(since 0.62).

[render]

The rendering (drawing) section controls methods that DOSBox uses to improve the speed and quality of the graphics displayed on the screen. E.g. it can "forget" (skip) the every 3rd screen update (which will save time), or it can try to smooth out some of the coarse low-resolution graphics that was used on old displays, but which looks bad when shown on a modern, high-resolution screen.

frameskip = *nnn*

How many frames DOSBox skips before drawing one.

Default is **0**.

aspect = **true** | **false**

Do aspect correction. It only affects non-square pixel modes like VGA Mode 13h, which has a resolution of 320x200 pixels and is used by many DOS games (DOOM, etc). Recommended as such games were designed for 4:3 displays, and without aspect correction will look distorted and not as the developer intended.

Default is **false**.

scaler = **none** | **normal2x** | **normal3x** | **tv2x** | **tv3x** | **rgb2x** | **rgb3x** | **scan2x** | **scan3x** | **advname2x** | **advname3x** | **advinterp2x** | **advinterp3x** | **2xsai** | **super2xsai** | **supereagle** | **hq2x** | **hq3x**

Specifies which scaler is used to enlarge and enhance low resolution modes, BEFORE any additional scaling done according to the Fullresolution and Windowresolution settings under [sdl]. To see comparisons between the different scalers, see Scaler.

none: no scaling is performed.

normal: nearest-neighbour scaling (big square pixels).

scan: like 'normal', but with horizontal black lines.

tv: like 'scan', but with darkened versions of the data instead of black lines.

rgb: simulates the phosphors on a dot trio CRT.

advname: smooths corners and removes jaggies from diagonal lines.

advinterp: identical to 'advname'.

sai: similar to 'advname' but with much softer color gradients and edges.

supersai: similar to 'sai' but sharper.

hq: a 'high quality' scaler which delivers a cleaner and sharper image than 'advname' or 'sai' scalers.

Default is **normal2x**.

Supported scalers which still need descriptions: supereagle

[cpu]

The CPU section controls how DOSBox tries to emulate the CPU, how fast the emulation should be, and to adjust it. DOSBox offers 4 different methods of CPU emulation.

core = simple | normal | dynamic | auto

CPU core used in emulation. The choices result in a different efficiency of DOSBox and in very rare cases have an effect on stability.

normal	The program is interpreted instruction by instruction. This approach is a lot more CPU demanding than dynamic core but allows for a more fine-grained time emulation and is needed on platforms for which DOSBox doesn't have a dynamic core.
simple	Basically the same as normal, but optimized for real-mode (older) games. In case a protected-mode game is started, it automatically switches back to normal core.
dynamic	The program instructions are, in blocks, translated to host processor instructions that execute directly. See also [3] (http://en.wikipedia.org/wiki/Just-in-time_compilation). In the most cases this approach is more efficient than interpretation, except for programs that employ massive self-modifying code. This option is not present on all host platforms.
auto	Real-mode programs are run with the normal core. For protected mode programs it switches to dynamic core, if available.
full	Deprecated.

Default is **auto**.

cputype = auto | 386 | 386_slow | 486_slow | pentium_slow | 386_prefetch

CPU Type used in emulation. **auto** is the fastest choice.

(since 0.73)

Default is **auto**.

cycles = fixed *nnnn* | max [*default%*] [*limit cycle limit*] | auto [*realmode default*] [*protected mode default%*] [*limit "cycle limit"*]

Amount of instructions DOSBox tries to emulate each millisecond. Set to **max** to automatically run as many cycles as possible. **auto** setting switches to **max** if appropriate.

fixed <i>nnnn</i>	Sets the emulated CPU speed to a fixed amount of cycles (<i>nnnn</i>). A value of 3000 equates 3 MIPS. If this value is too high some games will run too fast or crash. How high you can go depends on the power of your host CPU and on the selected core (above). If the value is too high for your CPU the emulation will slow down and the sound starts to skip.
max	Automatically sets the cycles so approximately the (optional) <i>default%</i> -value of your host CPU is used. If the value is not specified it defaults to 100%. The optional <i>limit</i> parameter limits the maximum speed to the specified value.
auto	For realmode games, this option switches to <i>realmode default</i> number of cycles or 3000 if not specified. When switching to protected mode, cycles is internally switched to max using the remaining optional parameters.

Default is **auto**.

Examples:

- *cycles=fixed 5000, cycles=5000* - All games you start are run with a fixed speed of ~5 MIPS. Useful for speed sensitive games or games that need a continuous CPU speed. You can change the actual value with Ctrl+F11 and Ctrl+F12 (keycombo) while DOSBox runs.
- *cycles=max* - All games you start run at the maximum speed your CPU permits. Use Ctrl+F11 and Ctrl+F12 to change the percentage of your CPU to be used.
- *cycles=max limit 50000* - All games you start run at up to 50000 cycles, depending on the power of your CPU.
- *cycles=max 50%* - About 50% of your CPU power will be used.
- *cycles=auto* - Realmode games will run at 3000 cycles. Protected mode games run with *cycles=max*.
- *cycles=auto 5000 50% limit 50000* - Realmode games run with 5000 fixed cycles, protected mode games with *cycles=max 50% limit 50000*.

cycleup = nnn

Amount of cycles to increase with keycombo.

Default is **10**. Setting it lower than 100 will be a percentage of the current value.

cycledown = nnn

Amount of cycles to decrease with keycombo.

Default is **20**. Setting it lower than 100 will be a percentage of the current value.

[mixer]

Configuration:Mixer(see Sound)

Here you can define the quality of emulated audio.

[midi]

(see Sound)

Here you can define any MIDI related settings. The term MIDI is commonly used to refer to background music found in games, but specifically it refers to synthesizer audio (which can be passed directly from emulated games to modern hardware).

mpu401 = intelligent | uart | none

Specifies which type of MIDI Processing Unit to emulate.

intelligent: (from Wikipedia) The MPU-401 can work in two modes, normal mode and UART mode. "Normal mode" would provide the host system with an 8-track sequencer, MIDI clock output, SYNC 24 signal output, Tape Sync and a metronome; as a result of these features, it is often called "intelligent mode", whereas...

uart: ... this simply emulates UART mode, which reduces the MPU-401 to just relaying in-/outcoming MIDI data bytes.

none: MIDI is not emulated.

mididevice = **default** | **win32** | **alsa** | **oss** | **coreaudio** | **coremidi** | **none**

A slightly confusing config name, because this isn't so much which MIDI device to use as which MIDI **interface** to use. As DOSBox currently does not emulate MIDI, but instead passes it through to an interface that does give MIDI playback support, this setting tells DOSBox which interface to pass MIDI data through to.

default: The default system MIDI playback device is used.

win32: Win32 MIDI playback interface is used.

alsa: Linux's Advanced Linux Sound Architecture playback interface is used.

oss: Linux's Open Sound System playback interface is used.

coreaudio: MacOS X's framework to render the music through the built-in OS X synthesizer.

coremidi: MacOS X's framework to route MIDI commands to any device that has been configured in Audio MIDI Setup.

none: MIDI is disabled.

midiconfig = *id of MIDI device*

As used by the MIDI interface described above, this specifies the ID which identifies the particular MIDI device to playback MIDI on. Can be determined on Windows using **MIXER /LISTMIDI** or on Linux using **pmidi -l** in the console.

[sbaster]

Configuration:SBaster(see Sound)

Sound Blaster related settings.

[gus]

Configuration:GUS (see Sound)

Gravis Ultra Sound related settings.

[speaker]

Configuration:PCSpeaker (see Sound)

PC Speaker related settings.

[joystick]

Configuration:Joystick (see Sound)

Joystick related settings.

[serial]

serialX = device [parameter:value]

device can be: dummy | modem | nullmodem | directserial

parameter is: irq

value is:

1. for directserial: realport (required), rxdelay (optional).
2. for modem: listenport (optional).
3. for nullmodem: server, rxdelay, txdelay, telnet, usedtr, transparent, port, inhsocket (all optional).

Defaults:

serial1=dummy

serial2=dummy

serial3=disabled

serial4=disabled

An example of how to configure an actual serial port for I/O use:

```
serial1=directserial realport:com1
```

[dos]

xms=true/false

default "true"

ems=true/false

default "true"

umb=true/false

default "true"

keyboardlayout=auto/none/XY

default "auto" (since 0.73. Previously "none"), see KEYB for supported values (keyboard layout codes/ids)

[ipx]

ipx=true/false

default "false"

See Connectivity on how to use IPX once it's enabled.

[autoexec]

Here you can define the contents of the AUTOEXEC.BAT file (which is executed immediately after DOSBox is initialized). Thus any commands listed here will be performed each time DOSBox is used.

Recommendations

Different games will naturally work best with different configuration settings. Something to consider is to define a conf file for each game that will set the appropriate configurations and start the game for you. Then, create a shortcut such as DOSBox.exe -conf "DOSbox-GameName.conf" so that you can start your game in the least possible number of steps.

Retrieved from "<https://www.dosbox.com/wiki/index.php?title=Dosbox.conf&oldid=134628>"

-
- This page was last edited on 12 July 2015, at 19:28.